MetaCAN: Improving Generalizability of Few-shot Anomaly Detection with Meta-learning

Zhisheng Lv*
College of Computer Science and Technology, National University of Defense Technology
Changsha, China
lzs@nudt.edu.cn

Jianfeng Zhang*
College of Computer Science and Technology, National University of Defense Technology
Changsha, China
jfzhang@nudt.edu.cn

Songlei Jian†
College of Computer Science and Technology, National University of Defense Technology
Changsha, China
jiansonglei@nudt.edu.cn

Chenlin Huang†
College of Computer Science and Technology, National University of Defense Technology
Changsha, China
clhuang@nudt.edu.cn

Hongguang Zhang
32010 Unit, PLA
Beijing, China
zhang.hongguang@outlook.com

Guansong Pang
School of Computing and Information Systems, Singapore Management University
Singapore, Singapore
pangguansong@gmail.com

Zhong Liu
College of Computer Science and Technology, National University of Defense Technology
Changsha, China
zhongliu@nudt.edu.cn

Abstract

Few-shot Anomaly Detection (AD) for images aims to detect anomalies with few-shot normal samples from the target dataset. It is a crucial task when only few samples can be obtained, and it is challenging since it needs to be generalized to different domains. Existing methods try to enhance the generalizability of AD by incorporating large vision-language models (LVLMs). However, how to transform category semantic information in LVLMs into anomaly information to improve the generalizability of AD remains a challenge facing existing methods. To address the challenge, we propose a few-shot AD method called MetaCAN, a novel category-to-anomaly network trained with AD meta-learning scheme based on an LVLM. Specifically, MetaCAN constructs the auxiliary training data and multiple tasks based on different categories to perform AD meta-learning, which ensures that the optimization toward the achievement of optimal anomaly detection across all categories. Moreover, MetaCAN introduces an image-image anomaly discriminator and an image-text anomaly detector to fully exploit the powerful multimodal semantic representations during auxiliary training. Once trained on auxiliary datasets, MetaCAN can be applied directly to other target datasets without retraining. Extensive experiments on six real-world datasets demonstrate that MetaCAN achieves state-of-the-art performance on cross-domain and cross-category anomaly detection tasks compared with existing methods.

CCS Concepts

• **Computing methodologies** → **Computer vision**.

Keywords

Anomaly Detection, Meta Learning, Few-shot Learning

ACM Reference Format:
Zhisheng Lv, Jianfeng Zhang, Songlei Jian, Chenlin Huang, Hongguang Zhang, Guansong Pang, and Zhong Liu. 2025. MetaCAN: Improving Generalizability of Few-shot Anomaly Detection with Meta-learning. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746252.3761253

1 Introduction

Anomaly detection (AD) plays a crucial role in human daily life. It is widely applied in various fields, such as industrial inspection, medical analysis, and security monitoring [29, 41]. Traditional research [1, 3, 8, 25, 27] typically proposes a single AD method for a specific category. Although these methods achieve good results, they are limited to specific categories and require a large amount of training data. These methods are impractical in real-world scenarios because abnormal samples are difficult to obtain. Therefore,

*Equal contributions.
†Corresponding authors.

CIKM '25, Seoul, Republic of Korea.

ACM ISBN 979-8-4007-2040-6/2025/11
https://doi.org/10.1145/3746252.3761253

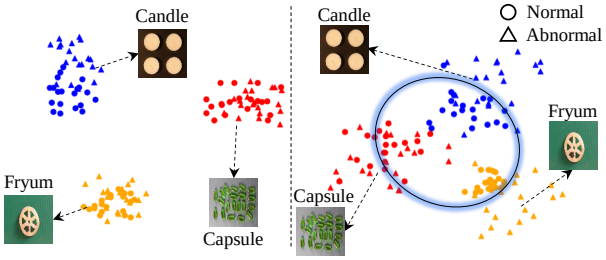


Figure 1: The t-SNE comparison of CLIP and MetaCAN on VisA. Left: CLIP's anomaly boundary is difficult to determine due to that different classes are grouped together. Right: After balancing the category and anomaly, MetaCAN get a clear boundary.

few-shot AD becomes a crucial task in such scenarios, which can detect cross-category anomalies with a few normal samples.

To achieve the goal of detecting cross-category anomalies with a few normal samples, existing work attempts from various aspects such as feature extraction and anomaly learning. RegAD [12] first proposes a category-agnostic few-shot AD approach through image alignment. This research marks the beginning of research on few-shot AD. Recently, large vision-language models (LVLMs) such as CLIP [30], have demonstrated strong few-shot capabilities in various tasks, including few-shot AD. For example, WinCLIP [14]introduces a windows-based CLIP, which employs prompt ensemble and window features to improve the performance of few-shot AD in industrial inspection. InCTRL [43] employs residual learning to create a general AD model, which can detect anomalies from different domain. IIPAD [26] learns a class-shared prompt generator to detect anomalies across multiple categories.

Although these methods have achieved some progress, they fail to consider the impact of category semantics in LVLMs, such as CLIP, on few-shot AD. As shown in Figure 1, in the original feature distribution of CLIP, the objects are clustered based on different category semantics, making it difficult to establish an effective decision boundary to distinguish between normal and abnormal samples. This is because LVLMs such as CLIP primarily align images with their corresponding category semantics, rather than the abnormality/normality in the images. As a result, CLIP has strong capabilities in few-shot classification based on category semantics, rather than in few-shot AD based on anomaly semantics. Directly using LVLMs such as CLIP does not yield a few-shot AD model with strong generalizability. Therefore, they still face a main challenge in building an effective few-shot AD method for various categories in different domains. *The main challenge is how to effectively transform the category semantic information in LVLMs into anomalous information to improve the generalizability of few-shot AD.*

To address the above challenge, and inspired by meta-learning methods for few-shot classification [10, 39, 40], we propose a few-shot AD method, named MetaCAN, which is a category-to-anomaly network trained using a meta-learning scheme tailored for AD (i.e., AD meta-learning). MetaCAN learns cross-category anomaly information with AD meta-learning and transforms the large vision-language model's powerful few-shot category classification capability into AD capability by the category-to-anomaly network, which enables the model to balance between category and anomaly, as shown in Figure 1. AD meta-learning constructs a new training scheme, including task and data constructions, which divides the auxiliary training set into N training tasks according to object categories, with each task containing a support dataset and a query dataset. Different from traditional meta-learning for classification, in AD meta-learning both support and query datasets contain query samples and k-shot reference samples. The support dataset and query dataset are used for the model's outer and inner updates, respectively. The total loss of each task is used to update the model parameters in the category-to-anomaly network. This process ensures that the optimization target of the model becomes the simultaneous achievement of optimal AD across all categories and enables direct AD on new data without fine-tuning. By leveraging anomaly learning across various feature levels (feature pyramids) and modalities (image-image, image-text), MetaCAN transforms the category semantic representation capability of LVLMs for AD.

In summary, our contributions are as follows.

- We propose a novel few-shot AD scheme, termed AD meta-learning, which includes new mechanisms for data construction, task construction, and parameter updating. This scheme improves the generalizability of few-shot AD across various categories and domains.
- We propose a few-shot AD method, i.e., MetaCAN, which is a category-to-anomaly network trained with the AD meta-learning scheme, that can effectively transform the large vision-language model's powerful semantic learning capability into AD ability.
- Comprehensive experiments are conducted on six datasets from different domains, demonstrating MetaCAN's strong few-shot AD capabilities across various categories over different domains.

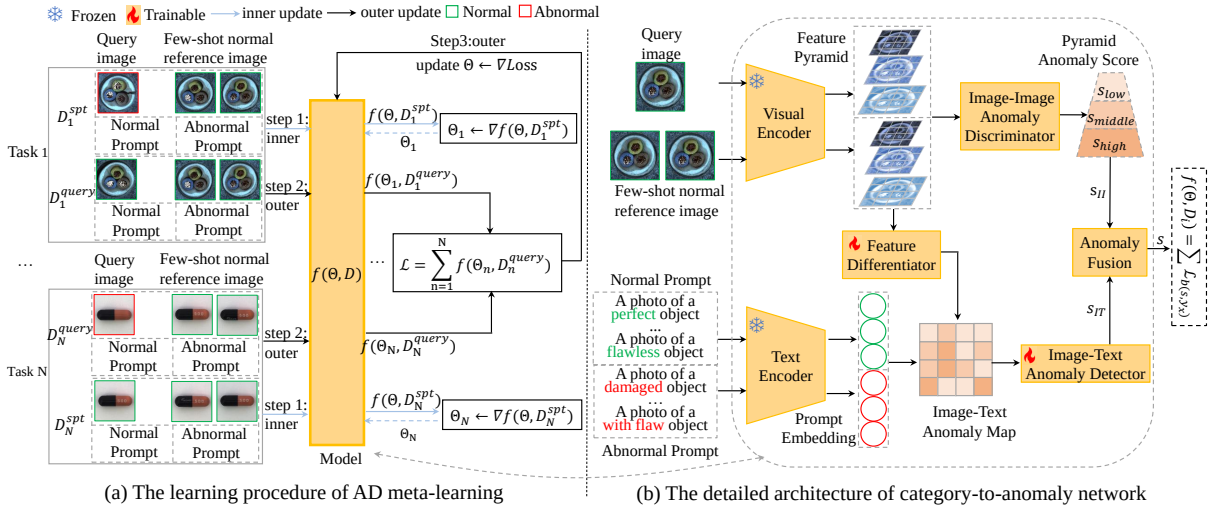
2 Related Work

2.1 Deep Anomaly Detection

Deep anomaly detection (AD), which addresses AD problems using deep neural networks. In the past, researchers have often focused on AD with a one-model-per-category approach. For example, some methods achieve AD by modeling normal states [1, 12, 21, 33], while others extract features using pre-trained neural networks [5, 6, 31], followed by anomaly classification. Although these methods have shown good performance on certain datasets, they follow a one-model-per-category approach, requiring retraining for each new category of AD. To address this issue, some researchers have begun exploring ways to adapt to new categories of AD without retraining. For example, PaDim [6] and PatchCore [31] employ metric learning to measure the distance between normal and abnormal samples for anomaly classification. while RegAD [12] uses the Siamese Network to compare samples with normal samples, achieving category-agnostic few-shot AD.

2.2 Large Vision-Language Models and AD

Recent large vision-language models (LVLMs) such as CLIP [30] have achieved significant success. With the emergence of various LVLMs [17, 18, 20], researchers began to explore their application



(a) The learning procedure of AD meta-learning (b) The detailed architecture of category-to-anomaly network

Figure 2: Overview of training and network structure of MetaCAN

in specialized fields [9, 24, 38], including AD. The AD methods can be categorized into zero-shot and few-shot. Zero-shot AD include AnomalyCLIP [42], Adaclip[4], etc., while few-shot AD include Winclip [14], AnomalyGPT [11], InCTRL [43], IIPAD [26], KAG-Prompt [36], and PromptAd[19]. Since our approach focuses on few-shot, we do not compare it with zero-shot AD. Among these few-shot AD, AnomalyGPT achieves strong performance by leveraging the PandaGPT [35] which demands substantial computational resources. Additionally, the PromptAd and KAG-Prompt focus on training a model for a class, which differs from our setting. In contrast, other methods still lack robustness for cross-category AD.

2.3 Meta Learning

Meta-learning is learning to learn. By acquiring sufficient knowledge, a model can quickly adapt to new tasks. It can be categorized into optimization-based and metric-based methods. Notable optimization-based methods include MAML [10] and Reptile [28]. MAML constructs various training tasks, each consisting of a support set and a query set, enabling the model to undergo meta-learning training. This allows the model to quickly adapt to the classification of different object categories. In metric-based learning, prominent works include MatchingNet [37] and SiamenseNet [15]. Matching Network incorporates external memory to enhance the network's learning capacity. However, these approaches are designed for category classification and cannot be directly applied to AD. This is because that object's categories are diverse and learning prior knowledge from other categories can help extend to new category classifications, AD only involves two classes: normal and abnormal. It is not possible to directly derive sufficient prior knowledge from these two classes for AD. Consequently, applying meta-learning to few-shot AD is still challenging.

3 Methods

To enhance the generalizability of few-shot AD, we propose MetaCAN. In this section, we will introduce the framework of MetaCAN, and explain how it improves the robustness of AD across different categories.

3.1 Overview

MetaCAN consists of two main components, i.e., AD meta-learning and category-to-anomaly network. As shown in Figure 2 (a), the AD meta-learning controls the task construction, data construction, and parameter updating procedures. Data constructed by AD meta-learning is then fed into the category-to-anomaly network, where it undergoes feature extraction and learning of anomaly information, resulting in anomaly scores. These scores are subsequently used by AD meta-learning to calculate the loss and update the model parameters. As shown in Figure 2 (b), the category-to-anomaly network extracts image and text features of the query samples and k-shot normal reference samples by feature encoders. In our work, we use CLIP as the visual and text encoders. The category-to-anomaly network constructs an image-image anomaly discriminator based on the feature pyramid and an image-text anomaly detector based on a query-reference feature differentiator, to detect anomalous information from the multimodal semantic information.

3.2 Task Construction in AD Meta-Learning

Similar to most few-shot learning methods, MetaCAN is first trained on an auxiliary dataset and then evaluates it on a separate test dataset. The key difference is that our approach selects k-shot samples from the test set as prompts without requiring any retraining. In contrast, other methods use the k-shot samples for fine-tuning. The comparison of data construction between existing few-shot AD methods (i.e., WinCLIP and InCTRL) and AD meta-learning is shown in Table 1. Before training, we first constructed the training task based on the auxiliary training dataset. To ensure that the model considers the optimization directions for different categories of AD during training, AD meta-learning divides the auxiliary dataset $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$ into N tasks according to the N types of objects, denoted as $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}$. Each task $\mathcal{T}_i$ is split into a

Table 1: Comparison of data construction between existing methods and AD meta-learning

Methods	Sample methods	Training data		
Existing methods	Random sample	Query sample x		
		Few-shot reference samples x^r		
		⋮		
		Query sample x		
		Few-shot reference samples x^r		
AD meta-learning	Construct tasks by category	Task 1 $\mathcal{T}_1$ (category 1)	Support set D_1^{spt}	Query sample x
				Few-shot reference samples x^r
			Query set D_1^{query}	Query sample x
				Few-shot reference samples x^r
		⋮	⋮	⋮
		Task N $\mathcal{T}_N$ (category N)	Support set D_N^{spt}	Query sample x
				Few-shot reference samples x^r
			Query set D_N^{query}	Query sample x
				Few-shot reference samples x^r

support set $\mathcal{D}_i^{spt}$ and a query set $\mathcal{D}_i^{query}$, with $\mathcal{D}_i^{spt}$ used for inner updates and $\mathcal{D}_i^{query}$ used for outer updates. Our objective is to capture anomaly information, not semantic information. To facilitate this, both the support and query sets include query samples x and k reference samples x^r, (known as k-shot), along with normal and abnormal text prompts.

3.3 Category-to-Anomaly Network

The category-to-anomaly network aims to capture abnormal information in the multimodal representations derived from the CLIP. It primarily consists of an image-image anomaly discriminator and an image-text anomaly detector to learn anomalies from multiple modality (i.e., image-image, image-text).

Image-Image Anomaly Discriminator To capture richer anomaly information, inspired by the feature pyramid [22], we propose an image-image anomaly discriminator that extracts multi-level features to capture richer anomaly information, allowing it to fit different categories of AD. Specifically, we first utilize the visual encoder ViT of CLIP to extract low-level, mid-level, and high-level features for any query sample x and its k-shot normal reference samples x^r, resulting in the feature pyramid $\mathcal{E} = \{\mathbf{E}_{low}, \mathbf{E}_{middle}, \mathbf{E}_{high}\}$ for the query sample x and $\mathcal{E}^r = \{\mathbf{E}^r_{low}, \mathbf{E}^r_{middle}, \mathbf{E}^r_{high}\}$ for the reference samples x^r. These feature pyramids enrich the representation of both the query and reference samples, effectively increasing the exposure of anomalous information.

Once obtaining these richer representations, we input them into the image-image anomaly discriminator. The discriminator calculates the difference between the query and reference samples based on their three levels of features to measure the discrepancies among the query samples. We use low-level features as an example to illustrate how features are converted into anomaly information. The low-level features of the query sample and the concatenated k-shot reference samples are denoted as $\mathbf{E}_{low} \in \mathbb{R}^{L\times D}$ and $\mathbf{E}^r_{low} \in \mathbb{R}^{kL\times D}$, with L and D represent the number and dimension of the features respectively. For each feature $\mathbf{e} \in \mathbb{R}^{1\times D}$ in the features $\mathbf{E}_{low}$ of the query sample , we calculate its similarity with $\mathbf{E}^r_{low}$ to identify the k most similar features.

$$\mathbf{c} = topk\langle \mathbf{e}, \mathbf{E}^r_{low} \rangle, \tag{1}$$

where $\mathbf{c} = \{c_1, c_2, \ldots, c_k\}$ represents the cosine similarity between $\mathbf{e}$ of the query sample and the k-shot features in the reference sample that are most similar to $\mathbf{e}$. $\langle , \rangle$ denotes the cosine similarity function.

After obtaining the similarity scores $\mathbf{c}$ between the query and the reference samples, we convert them into anomaly scores for the feature $\mathbf{e}$ of query sample:

$$m = 0.5 \times (1 - \frac{1}{n}\sum_{j=1}^{k} c_j), \tag{2}$$

where c_j represents j-th element in $\mathbf{c}$. In this way, for $\mathbf{E}_{low} = \{\mathbf{e}_1, ..., \mathbf{e}_L\}$, we obtain an anomaly map $\mathbf{m}_{low} \in [0,1]^{1\times L}$, with $\{m_1, ..., m_L\}$.

Finally, we take the maximum of $\mathbf{m}_{low}$ as the anomaly score for the low-level features:

$$s_{low} = max(\mathbf{m}_{low}), \tag{3}$$

where s_{low} represents the low-level anomaly score of x, with higher values indicating a higher degree of anomaly.

Similarly, we obtain the anomaly scores s_{middle} and s_{high} for the mid-level and the high-level features respectively. Then we combine these three levels of features to get the final anomaly score:

$$s_{II} = s_{low} + s_{middle} + s_{high}. \tag{4}$$

Image-Text Anomaly Detector To obtain anomalies from image-text information, existing methods [14, 43] achieve AD by directly calculating the similarity between the class token and text embeddings. For example, their text prompt is "A photo of a perfect {category}", "A photo of a damaged {category}", where {category} can refer to different object types. However, both the class token and the text contain redundant category information. To avoid the excessive influence of category information on anomaly detection, we discard category information in image-text anomaly detector.

To ensure that the prompt text is not influenced by category, we replace the {category} with *object*. Our normal prompt = "A

photo of a perfect *object*, ..., A photo of a flawless *object*"; abnormal prompt = "A photo of a damaged *object*, ..., A photo of an *object* with flaws" (Detailed prompts are presented in Appendix A). We then use CLIP's text encoder to obtain the text embeddings $\mathbf{T}_a$ and $\mathbf{T}_n$ of the normal and abnormal prompts, respectively. Similarly, to ensure that the visual embeddings are not influenced by categories, we do not use the class tokens. Due to the fact that high-level features contain more detailed semantics and can be aligned with the text information, we compare the high-level features of the query sample with those of the reference samples by calculating the difference. This difference is then combined with the prompt text to capture the anomaly information of the query sample.

First, we input the high-level features $\mathbf{E}_{high}$ of the query sample and the high-level features set $\mathcal{E}^r_{high} = \{\mathbf{E}^{r1}_{high}, \ldots, \mathbf{E}^{rj}_{high}, \ldots, \mathbf{E}^{rk}_{high}\}$ of k-shot reference samples into the feature differentiator to learn the differential feature $\mathbf{D}$ for a query sample.

$$\mathbf{D} = \mathbf{W}_2^T(\mathbf{W}_1^T(\mathbf{E}_{high} \ominus \frac{1}{k}\sum_{j=1}^{k}(\mathbf{E}^{rj}_{high}))), \tag{5}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ is the learned multi-layer perceptron that maps the image differential features to the same dimension as the text embedding. $\ominus$ denotes element-wise subtraction.

Next, we align the image differential features $\mathbf{D}$ and text embedding $\mathbf{T}_a, \mathbf{T}_n$ to get an image-text anomaly map for a query sample:

$$\mathbf{m}_{IT} = \frac{exp\langle \mathbf{D}, \mathbf{T}_a \rangle}{exp\langle \mathbf{D}, \mathbf{T}_n \rangle + exp\langle \mathbf{D}, \mathbf{T}_a \rangle}, \tag{6}$$

where $\langle , \rangle$ denote cosine similarity function. $\mathbf{m}_{IT}$ is the probability map, representing the probability that the differential feature of the query sample corresponds to the abnormal text.

Finally, the image-text anomaly detector maps the image-text anomaly map to obtain the anomaly score s_{IT}:

$$s_{IT} = Sigmoid(\mathbf{W}_4^T(\mathbf{W}_3^T\mathbf{m}_{IT})), \tag{7}$$

where $\mathbf{W}_3$ and $\mathbf{W}_4$ are the learnable parameters of the networks.

3.4 Training Scheme in AD Meta-Learning

To ensure that the model considers the optimization direction for each category of AD during training, we propose an AD meta-learning training method. This approach optimizes the model for different types of AD simultaneously, thereby enhancing its robustness across all categories.

Both the support and query sets contain query samples x and reference samples x^r, as well as normal and abnormal prompt texts. By inputting this data into the category-to-anomaly model, we obtain the final anomaly score for a query sample x:

$$s = (s_{II} + s_{IT})/4, \tag{8}$$

where the divisor is 4 because s_{II} represents the anomaly scores across three levels, ensuring that the anomaly score of the image information constitutes the primary component of the final score.

Based on this score, we can get the loss for a set $\mathcal{D}_i$, which can be support set $\mathcal{D}_i^{spt}$ or query set $\mathcal{D}_i^{query}$:

$$\mathcal{L}_i = f(\Theta, \mathcal{D}_i) = \sum_{x \in \mathcal{D}_i} \mathcal{L}_b(s, y_x), \tag{9}$$

Algorithm 1 The AD meta-learning algorithm of MetaCAN

Input: $\mathcal{P}(\mathcal{T})$: task distribution, S: number of iterations, N: number of categories.
Parameter: The model parameters Θ
Output: Update parameters Θ

```
Let iter = 0.
while iter ≤ S do
    Sample training task 𝒯 = {𝒯_1, ..., 𝒯_N} from the task distribution 𝒫(𝒯).
    for i = 1, ..., N do
        Sample 𝒟_i^spt from 𝒯_i
        Compute the loss of 𝒟_i^spt according to Eq.9
        Inner update and obtain the Θ_i (cf. Eq.10):
        Θ_i = Θ − β∇f(Θ, 𝒟_i^spt)
        Sample 𝒟_i^query from 𝒯_i
        Compute the loss of 𝒟_i^query according to Eq.9
    end for
    Compute the overall loss (cf. Eq.11):
    ℒ = Σ_{i=1}^N f(Θ_i, 𝒟_i^query)
    Outer update model parameters θ (cf. Eq.12):
    Θ' = Θ − α∇ℒ
    iter = iter + 1
end while
```

where $\mathcal{L}_b$ is a combination of Binary classification loss and Focal loss [23], f denotes the network in MetaCAN, Θ denotes the model parameters, and y_x is the label for x.

Next, we introduce our AD meta-learning process, which consists of three steps, as shown in Algorithm 1. Firstly, we input the support set $\mathcal{D}_i^{spt}$ of each task into the model. We calculate the gradient based on the loss and perform an inner update to obtain Θ_i:

$$\Theta_i = \Theta - \beta\nabla f(\Theta, \mathcal{D}_i^{spt}), \tag{10}$$

where β is learning rate of inner update, Θ_i is the parameters of the model. Note that Θ_i does not actually update the parameters of the model but serves as a reference for $\mathcal{D}_i^{query}$.

Secondly, we input the query set $\mathcal{D}_i^{query}$ of each task into the model and calculate the loss for each task using the previously obtained Θ_i. The losses from the test sets $\mathcal{D}_i^{query}$ are accumulated.

$$\mathcal{L} = \sum_{i=1}^{N} f(\Theta_i, \mathcal{D}_i^{query}), \tag{11}$$

where N is the number of tasks. Θ_i is used as a reference for $\mathcal{D}^i_{query}$ to optimize the model that has already been updated once internally.

Thirdly, the accumulated loss from the test sets $\mathcal{D}_i^{query}$ is used for an outer update to truly update the model parameters Θ. This ensures that the gradient of each model optimization step simultaneously considers the AD updates across N categories.

$$\Theta' = \Theta - \alpha\nabla\mathcal{L}, \tag{12}$$

where α is learning rate of outer update, Θ' is the updated model parameters. The overall loss ensures that the optimization target becomes the simultaneous optimization of AD across each category.

Table 2: AUROC and AUPRC results (mean±std) on four real-world AD datasets under few-shot. The results are the average of 3 runs. Boldface and underlining indicate the best and second performance, respectively. MVTec AD → * represents using MVTec AD as an auxiliary training dataset and testing on * dataset

Set up	Methods	Industrial AD		Industrial AD		Medical AD		Semantic AD	
		MVTec AD → VisA		MVTec AD → ELPV		MVTec AD → HeadCT		MVTec AD → MNIST	
		AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
2-shot	PaDiM	0.680±0.042	0.719±0.02	0.594±0.083	0.707±0.058	0.595±0.036	0.876±0.017	-	-
	PatchCore	0.817±0.028	0.841±0.023	0.716±0.031	0.840±0.031	0.736±0.096	0.913±0.002	0.603±0.009	0.482±0.025
	RegAD	0.557±0.053	0.614±0.037	0.571±0.016	0.679±0.005	0.602±0.018	0.854±0.009	0.608±0.026	0.612±0.013
	WinCLIP	0.842±0.024	0.859±0.021	0.726±0.020	0.849±0.010	0.915±0.015	0.975±0.012	0.612±0.007	0.614±0.005
	InCTRL	0.858±0.022	0.877±0.016	0.839±0.003	0.913±0.008	0.929±0.025	0.981±0.013	0.632±0.000	0.618±0.012
	IIPAD	0.858±0.018	0.880±0.017	0.816±0.034	0.905±0.022	0.828±0.009	0.941±0.002	0.595±0.009	0.582±0.022
	Ours	**0.896±0.020**	**0.905±0.020**	**0.863±0.003**	**0.930±0.002**	**0.937±0.014**	**0.983±0.004**	**0.653±0.002**	**0.620±0.008**
4-shot	PaDiM	0.735±0.031	0.758±0.018	0.612±0.080	0.724±0.067	0.622±0.013	0.890±0.011	-	-
	PatchCore	0.843±0.025	0.860±0.016	0.756±0.073	0.871±0.042	0.805±0.006	0.941±0.009	0.497±0.044	0.504±0.025
	RegAD	0.574±0.042	0.628±0.034	0.596±0.040	0.688±0.018	0.522±0.050	0.810±0.028	0.596±0.075	0.522±0.085
	WinCLIP	0.858±0.025	0.875±0.023	0.754±0.009	0.864±0.004	0.912±0.003	0.974±0.002	0.632±0.004	0.611±0.011
	InCTRL	0.877±0.019	0.902±0.027	0.846±0.011	0.916±0.009	0.933±0.013	0.984±0.011	0.643±0.007	0.620±0.004
	IIPAD	0.877±0.012	0.891±0.010	0.826±0.034	0.914±0.021	0.856±0.021	0.951±0.010	0.621±0.011	0.619±0.004
	Ours	**0.905±0.011**	**0.916±0.012**	**0.870±0.007**	**0.934±0.004**	**0.943±0.002**	**0.984±0.001**	**0.681±0.005**	**0.628±0.008**
8-shot	PaDiM	0.768±0.032	0.781±0.024	0.724±0.017	0.798±0.014	0.661±0.039	0.896±0.009	-	-
	PatchCore	0.860±0.026	0.873±0.022	0.837±0.016	0.915±0.007	0.817±0.034	0.931±0.006	0.526±0.019	0.530±0.037
	RegAD	0.589±0.040	0.643±0.032	0.633±0.027	0.696±0.015	0.628±0.026	0.931±0.006	0.573±0.076	0.566±0.048
	WinCLIP	0.868±0.020	0.880±0.021	0.814±0.010	0.897±0.007	0.915±0.008	0.975±0.003	0.641±0.004	0.616±0.006
	InCTRL	0.887±0.021	0.904±0.025	0.872±0.013	0.926±0.006	0.936±0.008	0.985±0.005	0.646±0.003	0.622±0.008
	IIPAD	0.885±0.010	0.908±0.008	0.857±0.007	0.933±0.003	0.868±0.007	0.961±0.004	0.625±0.015	0.624±0.005
	Ours	**0.911±0.009**	**0.920±0.011**	**0.881±0.003**	**0.940±0.002**	**0.951±0.008**	**0.986±0.002**	**0.694±0.009**	**0.638±0.006**

3.5 Inference

During inference, given a test sample x_i, we randomly select k-shot normal samples as reference samples x_i^r from the test dataset. The test sample x_i, k-shot reference samples x_i^r, together with normal and abnormal prompt text, are simultaneously fed into the category-to-anomaly network to obtain s_{II} and s_{IT}. Finally, we obtain the final anomaly score s according to Eq.8.

4 Experiments

4.1 Experimental Setup

Datasets To verify the effectiveness of our method, we adopt six datasets (i.e., MVTec AD [2], VisA [44], ELPV [7], HeadCT [32], AITEX[34] and MNIST[16]) from different domains for experiments. The MVTec AD is an industrial AD. It contains 3,629 and 1,725 samples. The VisA is also an AD dataset, including 9,621 normal samples and 1,200 abnormal samples. The ELPV is a solar cell image AD dataset consisting of 2,624 grayscale images. The HeadCT is a medical diagnostic dataset comprising 100 normal samples and 100 abnormal samples. The AITEX dataset is a textile AD dataset, which contains 140 normal samples and 105 abnormal samples. The MNIST is a classic handwritten digit recognition dataset from 0 to 9. More details are presented in Appendix B.2

Evaluation Metrics Following existing AD methods, we use two popular metrics Area Under the Receiver Operating Characteristic (AUROC) and Area Under the Precision-Recall Curve (AUPRC) to evaluate AD performance.

Comparison Methods We compare MetaCAN with state-of-the-art approaches, including full-shot AD methods, few-shot AD methods, and few-shot AD methods based on CLIP. The comparison methods are as follows:

- PaDiM [6] uses a Convolutional Neural Network (CNN) to extract features for each patch, and then represents each patch with a Gaussian distribution to obtain the anomaly probability for each sample.
- PatchCore [31] is a traditional full-shot AD method. It primarily addresses the cold-start problem in AD, which trains the AD model by using only normal samples.
- RegAD [12] addresses the one-model-per-category limitation of traditional methods by implementing a category-agnostic few-shot AD method through image alignment.
- WinCLIP [14] is the first to explore the use of CLIP for zero-shot and few-shot AD, achieving good AD performance without training.
- InCTRL [43] utilizes CLIP's strong few-shot capabilities to propose a general AD model, achieving good AD on datasets from different domains with only a few-shot normal sample.
- IIPAD [26] learns a class-shared prompt generator to detect anomalies across multiple categories.

Table 3: AUROC and AUPRC results (mean±std) on two datasets under few-shot AD. Taking VisA as the auxiliary training dataset.

Set up	Methods	Industrial AD VisA → MVTec AD AUROC	Industrial AD VisA → MVTec AD AUPRC	Industrial AD VisA → AITEX AUROC	Industrial AD VisA → AITEX AUPRC
2-shot	PaDiM	0.785±0.025	0.890±0.015	0.553±0.071	0.298±0.028
	PatchCore	0.858±0.034	0.939±0.012	0.646±0.070	0.403±0.083
	RegAD	0.640±0.047	0.837±0.034	0.625±0.116	0.373±0.068
	WinCLIP	0.931±0.019	0.965±0.007	0.701±0.015	0.522±0.006
	InCTRL	0.940±0.015	0.969±0.004	0.739±0.037	0.475±0.046
	IIPAD	0.942±0.014	0.967±0.008	0.721±0.020	0.493±0.014
	Ours	**0.946±0.005**	**0.971±0.002**	**0.765±0.041**	**0.533±0.084**
4-shot	PaDiM	0.805±0.018	0.909±0.013	0.592±0.040	0.294±0.042
	PatchCore	0.885±0.026	0.950±0.013	0.609±0.065	0.348±0.018
	RegAD	0.663±0.032	0.846±0.026	0.630±0.061	0.337±0.035
	WinCLIP	0.940±0.021	0.968±0.008	0.715±0.010	0.532±0.006
	InCTRL	0.945±0.018	0.972±0.006	0.743±0.016	0.498±0.022
	IIPAD	0.951±0.012	0.973±0.006	0.732±0.013	0.525±0.023
	Ours	**0.955±0.006**	**0.977±0.004**	**0.774±0.048**	**0.534±0.056**
8-shot	PaDiM	0.820±0.016	0.927±0.012	0.612±0.011	0.297±0.009
	PatchCore	0.922±0.019	0.962±0.013	0.615±0.047	0.376±0.045
	RegAD	0.674±0.033	0.855±0.021	0.675±0.014	0.379±0.022
	WinCLIP	0.947±0.025	0.973±0.009	0.733±0.013	**0.544±0.004**
	InCTRL	0.953±0.013	0.977±0.006	0.766±0.016	0.499±0.006
	IIPAD	0.954±0.011	0.974±0.005	0.754±0.010	0.528±0.008
	Ours	**0.958±0.002**	**0.977±0.003**	**0.802±0.035**	0.540±0.050

Table 4: Quantitative evaluation of modules on VisA and ELPV under 2-shot setting, w/o represents without.

Methods	VisA AUROC	VisA AUPRC	ELPV AUROC	ELPV AUPRC
w/o Discriminator	0.779	0.807	0.800	0.889
w/o Detector	0.865	0.877	0.817	0.902
w/o AD meta-learning	0.856	0.857	0.819	0.899
MetaCAN(Ours)	**0.896**	**0.905**	**0.863**	**0.930**

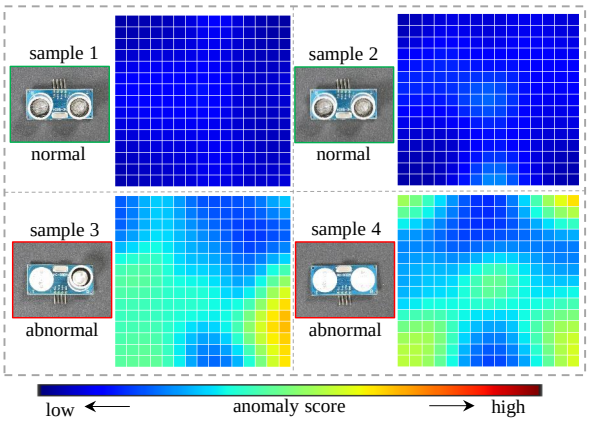


Figure 3: The generated image-text anomaly maps, whose dimensions are transformed from 1×225 to 15×15.

Implementation Details In the stage of feature encoding, we adopt CLIP as the backbone, with its visual encoder being the pre-trained ViT-B/16+ [13] and its language encoder being the pre-trained Transformer. We use Adam as the optimizer and set the initial learning rate α and β to 1e-3 by default. The auxiliary training dataset is based on MVTec AD or VisA. The batch size of the support and query sets are set to 32 and 8 respectively. The k-shot of reference samples is set to 2. The train epoch is set to 15. All settings apply to all comparison methods and the experimental results are from InCTRL's original paper [43]. Our method is implemented with Python 3.9 and Pytorch 2.1 framework. Detailed experimental setup is presented in Appendix B.1.

4.2 Comparison with SOTA methods

To verify the effectiveness of our method, we compare MetaCAN with the SOTA methods. Table 2 presents comparative results on four datasets when auxiliary training on MVTec AD. To evaluate the performance of MetaCAN with different auxiliary training datasets, Table 3 also presents comparative results on two datasets when auxiliary training on VisA.

The results of auxiliary training on MVTec AD. From Table 2, we can see that within the same domain (Industrial AD), MetaCAN consistently outperforms other methods under 2-shot, 4-shot, and 8-shot. Notably, MetaCAN shows the greatest improvement on the VisA dataset, with increases of 3.7%, 2.8% and 2.4% in AUROC, respectively, compared to the SOTA method. This shows that MetaCAN could achieve effective few-shot AD within the same domain. And MetaCAN has good generalizability of few-shot AD in industrial AD. In different domains (i.e., Medical AD and Semantic AD), MetaCAN demonstrates superior performance over other methods. Specifically, it achieves improvements of 2.1%, 3.8%, and 4.8% on the MNIST dataset under 2-shot, 4-shot, and 8-shot settings, respectively, compared to the SOTA method. This shows that MetaCAN is robust not only for cross-category AD within the Industrial AD domain but also for cross-domain AD with medical AD and semantic AD. And MetaCAN has good generalizability of few-shot AD in different domains.

The results of auxiliary training on Visa. Table 3 shows that MetaCAN improves AUROC performance on both the MVTec AD and AITEX datasets under 2-shot, 4-shot, and 8-shot settings compared to other methods when using VisA as an auxiliary training dataset. Specifically, on the AITEX dataset, MetaCAN achieves improvements of 2.5%, 3.1%, and 3.6% over the SOTA method under 2-shot, 4-shot, and 8-shot settings, respectively. This demonstrates that MetaCAN maintains robust anomaly detection capabilities across different categories, even when using diverse auxiliary datasets, indicating minimal performance degradation when the auxiliary dataset is altered.

4.3 Ablation Study

Quantitative Evaluation of Modules. To verify the effectiveness of each module in MetaCAN, we design an ablation study to explore the roles of the image-text anomaly detector (Detector), the

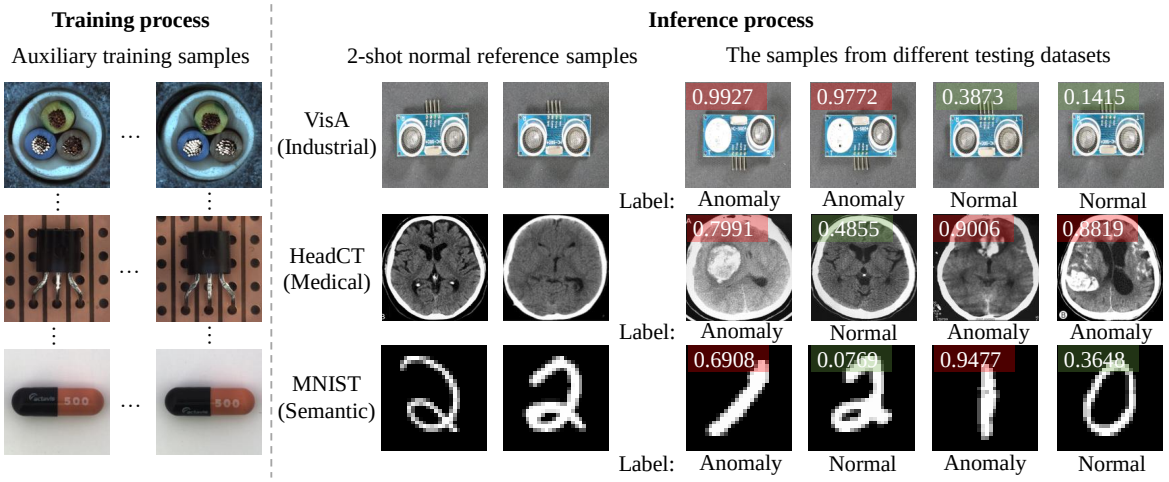


Figure 4: Case study from different domains. The results are obtained using MVTec AD as the auxiliary training dataset, the top and bottom of each image indicate the predicted scores and labels.

image-image anomaly discriminator (Discriminator), and AD meta-learning in MetaCAN. As shown in Table 4, regardless of whether the image-image anomaly discriminator, the image-text anomaly detector, or AD meta-learning is removed, MetaCAN's AUROC and AUPRC decrease on both the VisA and ELPV datasets. In comparison, when all modules are used together, MetaCAN achieves optimal performance. This indicates that both modules contribute individually, enhancing MetaCAN's overall effectiveness. Moreover, on the VisA dataset, the discriminator makes the greatest contribution in terms of performance, and AD meta-learning also plays a significant role. On the ELPV dataset, all three modules of MetaCAN contribute almost equally. This is because the three modules are designed to capture distinct types of information with varying emphases.

Visualization of Image-Text Anomaly Detector. To explore how the image-text anomaly detector operates within MetaCAN, we visualize the image-text anomaly map of two normal and abnormal examples from the VisA dataset. The results are shown in Figure 3. The objective of the image-text anomaly detector is to assign higher anomaly scores to abnormal samples compared to normal samples. Specifically, abnormal samples should yield anomaly maps with more high-value regions compared to normal samples. According to Figure 3, we can observe that the anomaly values in the image-text anomaly maps generated by two normal samples are generally low, whereas those generated by abnormal samples contain several regions with high anomaly values. This indicates that our image-text anomaly detector effectively distinguishes between normal and abnormal samples.

4.4 Case Study

To better illustrate the training and inference process of MetaCAN, we visualized the training and testing process on data from these three domains. As shown in Figure 4, MetaCAN employs MVTec AD as an auxiliary training dataset and then predicts anomaly scores for categories distinct from those in MVTec AD, using k-shot normal samples as references in the test dataset. For randomly selected test samples, MetaCAN achieves strong anomaly detection performance across all three domains. Under the prompt of 2-shot normal reference samples, MetaCAN assigns high anomaly scores to normal samples and low anomaly scores to abnormal samples across the three domains. This indicates that MetaCAN can effectively predict anomaly scores for samples from various domains with minimal reference data, demonstrating robustness in cross-domain and cross-categories few-shot AD.

5 Conclusions

In this paper, we propose a few-shot AD method, named MetaCAN, designed to enhance the generalizability of few-shot AD, to transform the category semantic information in the large vision-language models into anomalous information with AD meta-learning and category-to-anomaly network. By performing auxiliary training on a single training set, MetaCAN enables the model to detect anomalies of different categories over different domains in few-shot scenarios. During auxiliary training, MetaCAN employs an AD meta-learning training scheme, optimizing the category-to-anomaly network for different AD categories through both inner and outer updates. Additionally, the category-to-anomaly network uses an image-image anomaly discriminator and an image-text anomaly detector to transform the category semantic representation capability of the CLIP into anomaly detection capability. Experiments on six AD datasets from different domains demonstrate that MetaCAN achieves strong AD performance and enhances robustness across various AD categories and domains. In the future, we consider extending MetaCAN to more domains and modalities, e.g., video and text, not limited to images, by further unleashing the anomaly detection capabilities of large vision-language models.

Acknowledgments

The work is supported by the National Natural Science Foundation of China (No. 62172431, 62421002) and the Beijing Nova Program (No. 20220484139, 20240484747)

A. Detailed Prompt Text

Our prompt text is an improved version of WinCLIP's prompt text. To enhance the model's generalizability ability across categories and domains, we replaced the category in WinCLIP's prompt text with the term "object." Similar to WinCLIP [14], detailed normal prompt and abnormal prompt are shown is Table 5, our prompt text substitutes the [c] in the template_level text with the corresponding [c] from state_level, resulting in normal and anomaly prompt.

B. Detailed Experimental Setup

B.1 More Detailed Implementation

In this paper, we select CLIP as the backbone model. The visual encoder of CLIP is ViT-B/16+, and it consists of 12 layers of transformers. After encoding an image with ViT-B/16+, a length of 226 and a dimension of 896 embedding is obtained, consisting of one 'CLS' token and 225 patch tokens. The language encoder of CLIP is also a 12-layer pretrained transformer, producing a 640-dimensional embedding for each word. MetaCAN uses features extracted from the 1st, 7th, and 12th layers of ViT-B/16+ to form an image feature pyramid. All experiments are conducted on the Ubuntu 22.04.1 server with hardware including Intel(R) Xeon(R) Silver 4310 CPU @ 2.10GHz and 80G NVIDIA GPU A100. To evaluate the effectiveness of MetaCAN with different auxiliary training datasets, we conduct experiments in two configurations: (1) auxiliary training is conducted on MVTec AD, followed by testing on VisA, ELPV, HeadCT, and MNIST; (2) auxiliary training is conducted on VisA, followed by testing on MVTec AD and AITEX.

B.2 Datasets Details

We use six datasets (i.e. MVTec AD [2], VisA [44], ELPV [7], HeadCT [32], AITEX [34] and MNIST [16]) from different domains (i.e. Industrial AD, Medical AD and Semantic AD) for experiments.

- The MVTec AD dataset is an widely used industrial AD dataset. It contains a total of 5,354 samples, with 3,629 in the training set and 1,725 in the test set. There are 15 categories in total. And we use the test sets for evaluation.
- The VisA dataset is also a commonly used AD dataset, with a total of 10,821 samples, including 9,621 normal samples and 1,200 abnormal samples. There are 12 categories of objects. We only use the test set for evaluation.
- The ELPV dataset is a solar cell image AD dataset consisting of 2,624 grayscale images, including normal and abnormal solar cell images. We only use the test set for test.
- The AITEX dataset is a textile AD dataset that includes seven different textile structures. It contains 140 normal samples and 105 abnormal samples, all of which are 4096x256 pixel images. We use only the test dataset to eval.
- The HeadCT is a medical diagnostic dataset comprising 100 normal samples and 100 abnormal samples. Following the approach of InCTRL, we selected 25 normal samples and 100 abnormal samples as the test set.
- The MNIST is a classic handwritten digit recognition dataset consisting of 70,000 grayscale images representing ten digits, from 0 to 9. In this work, we set even digits as normal, and odd digits as anomalies.

Table 5: Detailed normal prompt and abnormal prompt

state_level (normal)	c := "object" c := "flawless object" c := "perfect object" c := "unblemished object" c := "object without flaw" c := "object without defect" c := "object without damage"
state_level (abnormal)	c := "damaged object" c := "object with flaw" c := "object with defect" c := "object with damage"
template_level	"a cropped photo of the [c]." "a cropped photo of a [c]." "a close-up photo of a [c]." "a close-up photo of the [c]." "a bright photo of a [c]." "a bright photo of the [c]." "a dark photo of a [c]." "a dark photo of the [c]." "a jpeg corrupted photo of a [c]." "a jpeg corrupted photo of the [c]." "a blurry photo of the [c]." "a blurry photo of a [c]." "a photo of the [c]." "a photo of a [c]." "a photo of a small [c]." "a photo of the small [c]." "a photo of a large [c]." "a photo of the large [c]." "a photo of a [c] for visual inspection." "a photo of the [c] for visual inspection." "a photo of a [c] for anomaly detection." "a photo of the [c] for anomaly detection."

Table 6: Complexity Comparison of Model

Method	Number of parameters	Inference Time (ms)
WinCLIP	0	510±3.6
InCTRL	334916	243±1.1
Our	1777615	262±1.2

B.3 Complexity of Model

To demonstrate the model parameters and inference speed of MetaCAN, we compare the parameter size and inference time per image of MetaCAN with those of WinCLIP and InCTRL. The comparison results are shown in Table 6. We can see that although our model has many more trainable parameters, our inference speed is significantly faster than WinCLIP, and only 20ms slower than InCTRL. However, our accuracy has been greatly improved compared to these two methods.

Disclosure of AI Tools

We only use GenAI tools for grammatical checking and writing polishing for this paper.

References

[1] Niamh Belton, Misgina Tsighe Hagos, Aonghus Lawlor, and Kathleen M Curran. 2023. Fewsome: One-class few shot anomaly detection with siamese networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2978–2987.

[2] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. 2019. MVTec AD–A comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9592–9600.

[3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. 2020. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4183–4192.

[4] Yunkang Cao, Jiangning Zhang, Luca Frittoli, Yuqi Cheng, Weiming Shen, and Giacomo Boracchi. 2024. Adaclip: Adapting clip with hybrid learnable prompts for zero-shot anomaly detection. In *European Conference on Computer Vision*. Springer, 55–72.

[5] Niv Cohen and Yedid Hoshen. 2020. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint arXiv:2005.02357* (2020).

[6] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. 2021. Padim: a patch distribution modeling framework for anomaly detection and localization. In *International Conference on Pattern Recognition*. Springer, 475–489.

[7] Sergiu Deitsch, Vincent Christlein, Stephan Berger, Claudia Buerhop-Lutz, Andreas Maier, Florian Gallwitz, and Christian Riess. 2019. Automatic classification of defective photovoltaic module cells in electroluminescence images. *Solar Energy* 185 (2019), 455–468.

[8] Hanqiu Deng and Xingyu Li. 2022. Anomaly detection via reverse distillation from one-class embedding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9737–9746.

[9] Aniket Deroy, Kripabandhu Ghosh, and Saptarshi Ghosh. 2024. Applicability of large language models and generative models for legal case judgement summarization. *Artificial Intelligence and Law* (2024), 1–44.

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*. PMLR, 1126–1135.

[11] Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. 2024. Anomalygpt: Detecting industrial anomalies using large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 1932–1940.

[12] Chaoqin Huang, Haoyan Guan, Aofan Jiang, Ya Zhang, Michael Spratling, and Yan-Feng Wang. 2022. Registration based few-shot anomaly detection. In *European Conference on Computer Vision*. Springer, 303–319.

[13] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. July 2021. OpenCLIP. *Zenodo* (July 2021).

[14] Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. 2023. Winclip: Zero-/few-shot anomaly classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19606–19616.

[15] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, Vol. 2. Lille, 1–30.

[16] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[17] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*. PMLR, 19730–19742.

[18] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*. PMLR, 12888–12900.

[19] Xiaofan Li, Zhizhong Zhang, Xin Tan, Chengwei Chen, Yanyun Qu, Yuan Xie, and Lizhuang Ma. 2024. Promptad: Learning prompts with only normal samples for few-shot anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16838–16848.

[20] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. 2023. Scaling language-image pre-training via masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 23390–23400.

[21] Jingyi Liao, Xun Xu, Manh Cuong Nguyen, Adam Goodge, and Chuan Sheng Foo. 2024. COFT-AD: COntrastive Fine-Tuning for Few-Shot Anomaly Detection. *IEEE Transactions on Image Processing* (2024).

[22] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2117–2125.

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[24] Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. 2021. Finbert: A pre-trained financial language representation model for financial text mining. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*. 4513–4519.

[25] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. 2023. Simplenet: A simple network for image anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20402–20411.

[26] Wenxi Lv, Qinliang Su, and Wenchao Xu. 2025. One-for-All Few-Shot Anomaly Detection via Instance-Induced Prompt Learning. In *The Thirteenth International Conference on Learning Representations*.

[27] Declan McIntosh and Alexandra Branzan Albu. 2023. Inter-realization channels: Unsupervised anomaly detection beyond one-class classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6285–6295.

[28] A Nichol. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999* (2018).

[29] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. 2021. Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)* 54, 2 (2021), 1–38.

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[31] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. 2022. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 14318–14328.

[32] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad H Rohban, and Hamid R Rabiee. 2021. Multiresolution knowledge distillation for anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 14902–14912.

[33] Eli Schwartz, Assaf Arbelle, Leonid Karlinsky, Sivan Harary, Florian Scheidegger, Sivan Doveh, and Raja Giryes. 2024. MAEDAY: MAE for few-and zero-shot AnomalY-Detection. *Computer Vision and Image Understanding* 241 (2024), 103958.

[34] Javier Silvestre-Blanes, Teresa Albero-Albero, Ignacio Miralles, Rubén Pérez-Llorens, and Jorge Moreno. 2019. A public fabric database for defect detection methods and results. *Autex Research Journal* 19, 4 (2019), 363–374.

[35] Yixuan Su, Tian Lan, Huayang Li, Jialu Xu, Yan Wang, and Deng Cai. 2023. Pandagpt: One model to instruction-follow them all. *arXiv preprint arXiv:2305.16355* (2023).

[36] Fenfang Tao, Guo-Sen Xie, Fang Zhao, and Xiangbo Shu. 2025. Kernel-Aware Graph Prompt Learning for Few-Shot Anomaly Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 7347–7355.

[37] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems* 29 (2016).

[38] Songhua Yang, Hanjie Zhao, Senbin Zhu, Guangyu Zhou, Hongfei Xu, Yuxiang Jia, and Hongying Zan. 2024. Zhongjing: Enhancing the chinese medical capabilities of large language model through expert feedback and real-world multi-turn dialogue. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 19368–19376.

[39] Huaxiu Yao, Linjun Zhang, and Chelsea Finn. 2022. Meta-Learning with Fewer Tasks through Task Interpolation. In *International Conference on Learning Representations*.

[40] Han-Jia Ye and Wei-Lun Chao. 2022. How to Train Your Maml to Excel in Few-shot Classification. In *International Conference on Learning Representations*.

[41] Zahra Zamanzadeh Darban, Geoffrey I Webb, Shirui Pan, Charu Aggarwal, and Mahsa Salehi. 2024. Deep learning for time series anomaly detection: A survey. *Comput. Surveys* 57, 1 (2024), 1–42.

[42] Qihang Zhou, Guansong Pang, Tian Yu, Shibo He, and Jiming Chen. 2024. Anomalyclip: Object-agnostic prompt learning for zero-shot anomaly detection. In *International Conference on Learning Representations*.

[43] Jiawen Zhu and Guansong Pang. 2024. Toward generalist anomaly detection via in-context residual learning with few-shot sample prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17826–17836.

[44] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. 2022. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *European Conference on Computer Vision*. Springer, 392–408.